EXHIBIT FFF

Solaris Containers — What They Are and How to Use Them

Menno Lageman, Sun Client Solutions

Sun BluePrints™ OnLine—May 2005

microsystems

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650 960-1300

Part No.819-2679-10
Revision 1.0, 4/28/05
Edition: May 2005

1

Executive Overview

Over the years businesses have been building large-scale information systems to solve business problems, with a focus on building scalable and highly available IT infrastructures that can adapt change. Providing sufficient availability and performance for business applications was the primary driver for these efforts. Today, the need to protect technology investments and provide the same service levels at a lower price point is shifting the focus to reducing IT infrastructure cost and improving end user service level management. Sun believes businesses can accomplish this goal by utilizing the facilities available in *Solaris Containers*.

End user application services are frequently comprised of components that are distributed across multiple servers. To reduce costs, businesses are eager to consolidate these applications onto fewer servers, but they must be careful to maintain isolation between applications. Hardware and software advances have led to the concept of *server virtualization*, the partitioning of network servers into several independent execution environments. Server virtualization allows a data center to be viewed and managed as a set of compute resources rather than a room of individual systems.

Solaris Containers

Sun's next advance in server virtualization is a concept called Solaris Containers. Solaris Containers provide isolation between software applications or services using flexible, software-defined boundaries. Applications can be managed independently of each other, even while running in the same instance of the Solaris™ Operating System (Solaris OS). Solaris Containers create an execution environment within a single instance of the Solaris OS and provide:

- *Full resource containment and control* for more predictable service levels
- *Software fault isolation* to minimize fault propagation and unplanned downtime
- *Security isolation* to prevent unauthorized access as well as unintentional intrusions

The primary benefits of Solaris Containers are:

- *Reduced management costs* through server consolidation, and a reduced number of operating system instances
- *Increased resource utilization* with dynamic resource reallocation between Containers
- *Increased service availability* by minimizing fault propagation and security violations between applications
- *Increased flexibility* because software based Containers can be dynamically reconfigured
- *Increased accuracy and flexibility of accounting*, based on workloads rather than systems or processes

How This Article is Organized

- Chapter 2, “Introduction,” provides an overview of the features of Solaris Containers.
- Chapter 3, “Workload Management,” describes the nature of workloads today and how resource management techniques can be used to gives those workloads the resources they need.
- Chapter 4, “Managing Workloads — An Example,” provides an example of how to use projects to manage workloads effectively.
- Chapter 5, “Dynamic Resource Pools,” describes how dynamic resource pools can be used to automatically allocate resources.
- Chapter 6, “Resource Pools — An Example,” provides an example of how to use resource pools to partition the CPU resources on a system.
- Chapter 7, “Solaris Zones,” describes the new Zones features in the Solaris 10 Operating System.
- Chapter 8, “Using Zones — An Example,” illustrates how to use Zones to create virtual environments on a system.
- Chapter 9, “Containers — An Example,” describes how to create Containers that can be used to consolidate multiple copies of applications onto a system.
- Chapter 10, “Summary,” provides links to more information.

Typographic Conventions

TABLE 1-1 describes the typographic conventions used in this article.

TABLE 1-1 Typographic Conventions

Typeface	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`%` **`su`** `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this.
AaBbCc123	Command-line placeholder text; replace with a real name or value	To delete a file, type `rm` *filename*.

2

Introduction

Today, businesses often design their systems with extra capacity to handle occasional peak loads to maximize revenue during periods of high demand. This extra system capacity remains unused during periods of normal demand. By allowing other applications to borrow this unused capacity a more cost-effective solution can be realized. During periods of high demand resources can be dynamically reallocated to important applications. Sharing resources in this way leads to higher resource utilization, reduces capital and system management costs by reducing the total number of systems required. For the consolidation of applications onto fewer systems to be effective, applications must be able to be managed independently. This requires the ability to control resource utilization, isolate faults, and manage security between multiple applications on the same server. In other words, it requires the establishment of virtual server boundaries within the server.

One of the first steps in this direction was the introduction of Dynamic System Domains on large Sun servers. With Dynamic System Domains, a server can be divided into several domains, each running its own copy of the Solaris OS. The domains provide hardware isolation between the applications so that faults in one domain do not propagate to applications in other domains. Domain boundaries can be dynamically partitioned to adapt to changing resource requirements. Resources can be moved from one domain to another without requiring a restart of the system. This adds flexibility to the data center while maintaining security and isolation from faults in other domains.

Starting with Solaris Resource Manager 1.x in the Solaris 2.6 OS, Sun has gradually enhanced the ability to control resource utilization and separate applications running in a single instance of the Solaris OS. Several technologies have been added to the Solaris OS over the years, providing additional capabilities and finer control over resource utilization. Examples of such technologies include the Solaris 9 Resource Manager and Resource Pools in the Solaris 9 OS. These technologies allow users to create a Solaris Container, an application or *service* that has one or more resource boundaries associated with it. These resource boundaries can limit CPU or memory consumption, network bandwidth, or even be a processor set. As a result, Solaris Containers are a prime enabler for server consolidation.

With the introduction of Solaris Zones in the Solaris 10 OS, Sun is taking Solaris Containers a step further by allowing servers to be partitioned in sub-CPU granularity. A Solaris Zone is a complete execution environment for a set of software services — a separate, virtual Solaris

environment within a Solaris instance. A Zone provides a virtual mapping from software services to platform resources, and allows application components to be isolated from each other even though they share a single Solaris OS instance. It establishes boundaries for resource consumption and provides isolation from other Zones on the same system. The boundaries can be changed dynamically to adapt to changing processing requirements of the applications running in the Zone.

Solaris Containers can be built using one or more the following technologies. These technologies can be combined to create Containers tailored for a specific server consolidation project.

- Solaris Resource Manager, for workload resource management
- Resource Pools, for partitioning
- Zones, for isolation, security and virtualization

It is important to note that a Solaris Container is not equivalent to a Solaris Zone. Zones technology can be used to create a Container with certain characteristics, such as the isolation provided by the virtual Solaris environment. But it is also possible to create another Solaris Container using Resource Pools technology if the required characterics of that Container can be met with the features Resource Pools provide. So while a Zone is a Container, a Container is not necessarily a Zone.

Workload Resource Management

One of the inhibitors for consolidating multiple applications onto a single server is the lack of control over the resources utilized by applications. Consider the example of a company that wants to consolidate two database servers onto one system to decrease the number of systems to manage, as well as the number of software licenses required.

The first database is used by an on-line sales application while the second database is used by a marketing application. Because the sales application supports the core business of the company, it should be guaranteed a certain minimum amount of CPU when needed. The marketing application is a supporting application, and the CPU requirements of the database server are much less stringent. Without a mechanism to enforce these requirements, these two

applications cannot be consolidated successfully onto one system. With Solaris Containers, these business requirements can be implemented by establishing the appropriate CPU resource boundaries using the *Fair Share Scheduler* (Figure 2-1).

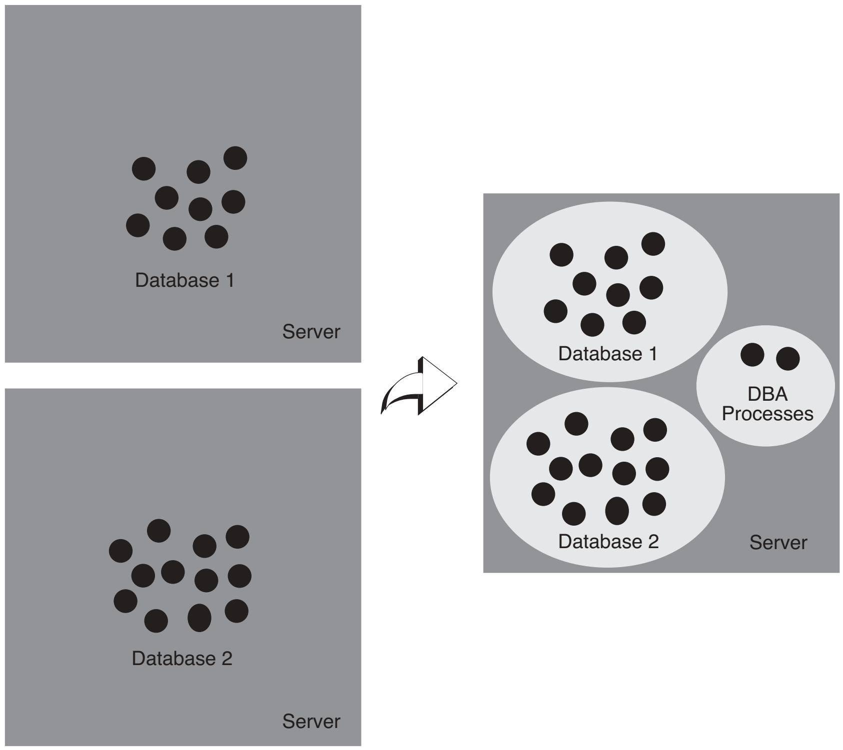

FIGURE 2-1 Solaris Containers provide an environment that fosters the safe consolidation of applications onto a single server

The Fair Share Scheduler controls the allocation of available CPU resources among workloads based on their relative importance. See Chapter 3 for more information on workload management and Chapter 4 for a hands-on example of workload management.